

2020-02-13

# An Exact Algorithm for Small-Cardinality Constrained Portfolio Optimisation

Bernik, Nataliia

<http://hdl.handle.net/10026.1/15298>

---

10.1080/01605682.2020.1718019

Journal of the Operational Research Society

Taylor & Francis

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*

ORIGINAL PAPER

## An Exact Algorithm for Small-Cardinality Constrained Portfolio Optimisation

David I. Graham<sup>a</sup> and Matthew. J Craven <sup>a</sup>

<sup>a</sup>Centre for Mathematical Sciences, University of Plymouth, Plymouth PL4 8AA, UK.

### ARTICLE HISTORY

Compiled January 13, 2020

### ABSTRACT

Real-world portfolio optimisation problems are often NP-hard, their efficient frontiers (EFs) in practice being calculated by randomised algorithms. In this work, a deterministic method of decomposition of EFs into a short sequence of sub-EFs is presented. These sub-EFs may be calculated by a quadratic programming algorithm, the collection of such sub-EFs then being subjected to a sifting process to produce the full EF. Full EFs of portfolio optimisation problems with small cardinality constraints are computed to a high resolution, providing a fast and practical alternative to randomised algorithms. The method may also be used with other practical classes of portfolio problems, complete with differing measures of risk. Finally, it is shown that the identified sub-EFs correspond closely to local optima of the objective function of a case study evolutionary algorithm.

### KEYWORDS

Portfolio selection; cardinality constraint; deterministic methods; quadratic programming; evolutionary algorithm; operational research

## 1. Introduction

The portfolio optimisation problem is a ubiquitous real-world problem in financial mathematics and combinatorial optimisation. Simply put, the problem states that given asset returns over some time period, an investor must decide upon which assets to invest in, and in which proportions, to maximise portfolio return or minimise portfolio risk. This is subject to certain constraints which often add to the computational complexity. In particular, the difficulty of the cardinality constrained portfolio optimisation problem - where the maximum number of assets that may be chosen is restricted - is on a spectrum. It is well-known that for large cardinalities the problem is equivalent to the unconstrained portfolio optimisation problem. This problem is solved via quadratic programming. Conversely, for small cardinalities and large numbers of assets the problem is NP-hard (Bach, Ahipasaoglu, & d'Aspremont, 2010; Mansini & Speranza, 1999; Moral-Escudero, Ruiz-Torrubiano, & Suárez, 2006). Problems with this level of difficulty may be efficiently approximated (rather than solved exactly) using metaheuristics such as evolutionary algorithms (EAs).

Work to approximate solutions to such problems via metaheuristics abound. For example, the seminal work of Chang, Meade, Beasley, and Sharaiha (2000) compared a

genetic algorithm, tabu search algorithm and simulated annealing on the cardinality-constrained problem with minimum asset proportions. In another example of this approach, Lwin, Qu, and Kendall (2014) used ‘MODEwAwL’ - a hybrid of differential evolution and machine learning - on the problem with additional constraints imposed and compared the approach with NSGA-II and other standard EAs. The same problems, and others, were examined by the work of Anagnostopoulos and Mamanis (2011), which compared five benchmark EAs (NPGA2, NSGA-II, PESA, SPEA2 and e-MOEA). Each of the above works used the OR-Library (Beasley, 1990) standard benchmark indices. There have been approaches using other kinds of datasets also (for example, the work of Duran, Cotta, and Fernández-Leiva (2012) (European mutual funds market), Lin and Liu (2008) (Taiwan mutual fund data), Mansini and Speranza (1999) (Milan stock exchange), and Craven and Graham (2019) (datasets derived from the S&P 500 index - more information is given in Section 4.1)).

However, EAs (more generally, metaheuristics) are not necessarily a panacea for such problems. Such methods are often complex and have a long runtime, exhibiting accuracy that increases with run time. Small cardinalities, in particular, typically result in a high EA computational cost due to the difficulty in finding ‘good’ solutions, indicating an alternative method may be more suitable for these cases. The work of Cesarone, Scozzari, and Tardella (2013) formulated a heuristic ‘increasing set’ algorithm to solve the problem, reporting efficient frontiers for the OR-Library benchmark instances (Beasley, 1990). There are also other examples of exact methods such as branch-and-bound (Gao & Li, 2013) and cutting-plane methods (Bertsimas & Cory-Wright, 2019). In the present work, an alternative, intuitively simpler approach is adopted that not only solves the portfolio optimisation problem but which also sheds light upon results obtained using evolutionary algorithms to solve such problems. Building upon the preliminary work of Craven and Graham (2017), ‘cardinality constrained’ efficient frontiers (CCEFs) of the portfolio optimisation problem are calculated. These EFs are constructed by reducing the problem to a sequence of quadratic programming problems, attempting to minimise the length of this sequence by identification of applicable non-dominated partial (sub-) EFs. The final constrained EF is then produced by a process of sifting from this pooled collection of sub-EFs (see Section 3 for details). Comparisons of the solutions with (modified) solutions of Cesarone et al. (2013) are provided. The contributions of this work are as follows:

- (1) The work provides an efficient and practical method of constructing CCEFs, each from a union of non-dominated sub-EFs, as benchmark optimal solutions for EAs. This is illustrated on the benchmark datasets of Beasley (1990) and two datasets produced from the S&P 500 index.
- (2) These CCEFs are calculated to a high accuracy, comparing well with Cesarone et al. (2013).
- (3) The work demonstrates a practical alternative to using metaheuristics such as EAs for small cardinality constraints in such problems, providing solutions to a higher degree of accuracy than those methods.
- (4) A visualisation of EA evolution and local objective function optima as part of a full solution is provided. This is used to, firstly, display explicitly where the final CCEFs come from and, secondly, in conjunction with the benchmark optimal solutions above, identify typical local optima locations of EAs and hence where such approaches are likely to struggle.
- (5) Finally, a substantial applied mathematical analysis of portfolio problems under the given constraints is provided.

The work proceeds thus. In Section 2, the formulation of the problem is briefly stated. An explanation and listing of the approach of this work is provided in Section 3, followed in Section 4 by results and observations of the approach applied to the discussed datasets. Numerous plots are presented - these plots show excellent correspondence of the CCEFs produced with those of Cesarone et al. (2013). Section 5 links the non-dominated sub-EFs produced by the current approach with local minima solutions produced by metaheuristics, giving an insight into properties of those solutions. This is followed in Section 6 by a comparison of the work with other methods. A conclusion is then provided in Section 7, highlighting the practicality of the approach, its generalisation and further use on associated optimisation problems.

## 2. Formulations of the Problem

In a financial market, suppose an investor wishes to invest in a set of assets of size  $n$ . As a proportion of their total investment, each asset  $i = 1, \dots, n$  is assigned a proportion or weight  $x_i$  giving a portfolio vector of weights  $\mathbf{x} = (x_1, \dots, x_n)$ . Short-selling is not allowed and thus the elements of this vector are assumed to be non-negative and sum to 1. Over a given period suppose that  $\mu_i$  is the expected return of asset  $i$  and  $\sigma_{ij}$  is the covariance between the returns of assets  $i$  and  $j$ . The expected return of the portfolio  $\mathbf{x}$  is given by

$$R = \sum_{i=1}^n \mu_i x_i, \quad (1)$$

with the variance (risk) being written as

$$V = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j. \quad (2)$$

The Markowitz problem (in this mean-variance model) of minimising portfolio risk  $V$  for a given return level  $R > R_0$  is a quadratic problem and so may be solved by quadratic programming (QP). However, to be more real-world relevant, traders commonly impose additional constraints (examples include cardinality constraints and minimum proportions). The cardinality constrained portfolio optimisation problem is when the investor wishes to restrict their investment to a subset of  $k \leq n$  assets. This is often done to mitigate transaction costs. The formulation that will be studied is the following one-objective formulation, an equivalent  $\lambda$ -parameterised version of the above two-objective formulation. The parameter  $\lambda$  represents aversion to risk. The formulation is stated as follows:

Find a portfolio  $\mathbf{x}$  such that  $\lambda V - (1 - \lambda)R$  is minimised subject to:

$$\begin{aligned} \sum_{i=1}^n x_i &= 1 \\ x_i &\geq 0 \text{ for } i = 1, \dots, n \end{aligned} \quad (3)$$

$$\#\{j : x_j \neq 0\} \leq K$$

Note that the final constraint is an inequality rather than an equality. This is the constraint followed by many practitioners (e.g., Chang et al. (2000)). It is well known that the problem is NP-hard, via various reductions (Mansini & Speranza, 1999; Moral-Escudero et al., 2006) and thus analytical solutions are not available in many practical situations. Finding an approximate solution to the problem is efficiently done with a metaheuristic algorithm (Anagnostopoulos & Mamanis, 2011; Chang et al., 2000; Craven & Graham, 2019; Lwin et al., 2014). Typically, the longer such algorithms are run, the more (asymptotically) accurate the approximation. The Cardinality Constrained Efficient Frontier (CCEF) is the set of all optimal solutions to (3). It is non-quadratic and, in fact, highly unpredictable for certain ranges of returns Cesarone et al. (2013). The above formulation allows parameterisation of the CCEF by  $\lambda \in [a, b] \subseteq [0, 1]$  (with  $\lambda = 0$  being the point at which return is maximised and  $\lambda = 1$  the point at which risk is minimised). However, in some cases, e.g., for small  $K$ , it will be shown that a good approximation to the CCEF may be found in an analytic manner. The next section describes the method.

### 3. The Algorithm

#### 3.1. Explanation

In finding the EF for a subset of  $K$  assets from  $n$  using QP techniques, each of the  $\binom{n}{K}$  possible combinations of assets is considered separately. Even for moderate  $n$ , this can represent a very large number of cases. In principle, the full CCEF for each of these cases (i. e., for each value of  $\lambda$ ) must be found and then the best (risk, return) combination for each  $\lambda_i$  identified by collating information at the end of the computation. For large datasets, this would be prohibitively costly and most of the information generated would be jettisoned at the end of the calculation. To make such computations feasible, it has been necessary to identify and eliminate ‘dominated’ subsets, where the CCEF generated from this subset would lie ‘below and to the right of’ previously-computed CCEFs and would therefore not contribute to the overall EF. By re-ordering the data in order of decreasing return (with corresponding re-ordering of the covariance matrix) the vast majority of subsets are indeed dominated, allowing very large overall numbers of cases to be computed. This re-ordering is essential for efficient removal of dominated cases.

This efficient removal is performed by consideration and exploitation of dominance (building upon the preliminary work of Craven and Graham (2017)). Dominance is defined intuitively as follows. Figure 1 (a) (left) shows curve  $C_2$  (the ‘test EF’) and a potentially dominating curve  $C_1$ . Observe that the grey triangle emitting from a point on  $C_2$  to two points  $C_1$  has ‘positive area’ (i.e., the triangle emits in a north-west direction). If this holds for every point on  $C_2$  and every pair of distinct points on  $C_1$  then  $C_2$  is said to be dominated by  $C_1$ . On the contrary, on the right of the same subfigure,  $C_2$  is not dominated by  $C_1$  at all points since there is a triangle emitting from a point of  $C_2$  with ‘negative area’. Hence  $C_2$  is non-dominated; indeed, it can be seen that the portion of  $C_1$  lying above  $C_2$  and the majority of  $C_2$  lying to the left of  $C_1$  gives the non-dominated front. Subfigure (b) shows a sample of dominated sub-EFs in red, with some final (dominating) CCEF points in green.

The formulation requires constrained minimisation of the risk. The risk is a quadratic form as a function of the asset weights and the constraints are linear. Such problems (i.e. minimisation of quadratic form subject to linear constraints) can be

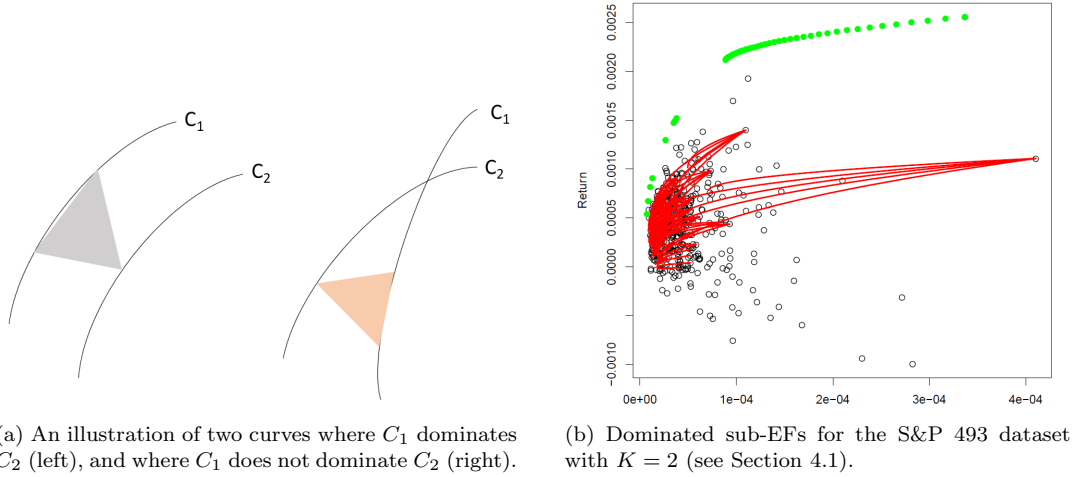


Figure 1.: Illustrations of dominance.

solved using quadratic programming. An overview of the algorithm follows.

Firstly, create the EF for the sub-portfolio of renumbered assets  $1, 2, \dots, k$ . Then test whether the sub-EF generated by a given sub-portfolio would be dominated. This is done by calculating three points on the ‘sub-EF’. The first point gives the maximum return  $R_{\max}$ , with corresponding risk  $V_{\max}$  (corresponding to  $\lambda = 0$ ).  $R_{\max}$  and  $V_{\max}$  are provided as input data. A second point minimises the risk  $V_{\min}$  (at  $\lambda = 1$ ) and has associated return  $R_{\min}$ .  $R_{\min}$  and  $V_{\min}$  are calculated using QP. For the sake of efficiency, rather than compute the whole sub-EF, instead choose a third point with coordinate  $(\frac{1}{2}(V_{\max} + V_{\min}), \frac{1}{4}(3R_{\min} + R_{\max}))$ . This assumes that the sub-EF is closely approximated by a quadratic and is exact for  $K = 2$ . The choice of this ‘third point’ is discussed further in Section 4.4. It is then tested whether any of these three points is not dominated. A point is dominated if the gradient of the line joining that point to any point  $(V_i, R_i)$  (where  $i = 1, \dots, n_\lambda$ ) on the current full EF is greater than the gradient of the current full EF at that point. Note that vector products are used to evaluate this condition in the algorithm. Its effectiveness relies upon the fact that, in the cost-minimisation version of the Markowitz problem, the full EF is a concave function (see, e.g., (Chang et al., 2000, pp. 1274-1275), Craven and Graham (2019)). The test EF is not dominated if any of the three test points is not dominated in this way, and is dominated otherwise. If the test EF is not dominated then calculate, via the R function `Solve.QP`, and store the full sub-EF for this sub-portfolio. The collection of all computed full sub-EFs will then be sifted at the conclusion of the computation.

### 3.2. Listing

The method is simple and effective for lower cardinalities (e.g.,  $K = 2$  or  $K = 3$ ) since the number of combinations (sub-portfolios) that are checked by the algorithm is manageably small. The method also provides solutions to a high degree of accuracy. The method consists of two distinct phases. In the first phase, non-dominated sub-EFs are identified as described above. This can be described as the elimination of ‘curvewise-dominated’ sub-EFs. In the second phase the sub-EFs are collated and sifted by identifying and eliminating individual points as described below. Sifting starts

with the point of highest return  $R$  and sifts out all points ‘pointwise dominated’ by this point (here, point (A) pointwise dominates point (B) iff both  $R(A) \geq R(B)$  and  $V(A) \leq R(B)$ ). From the remaining points, find the point with the next highest return and sift out any points dominated by it. This process is repeated until all pointwise dominated points are eliminated.

---

**Algorithm 1** QP Algorithm for the One-Variable Formulation

---

**input:** returns vector  $R(1, \dots, n)$  and covariance matrix  $\sigma_{n \times n}$ ;  
total number of assets,  $n$ , and in sub-portfolio,  $K$ , parameter ‘ $tol$ ’.

**output:**  $T = \{(V_j, R_j) : j = 1, \dots, n_\lambda\}$ , the set of risk-return pairs on the EF;  
mincost( $\lambda_j$ ): minimum cost for points on EF;  $n_{nd}$ : the number of non-dominated sub-EFs;  
 $S = \{(V_{i,j}, R_{i,j}) : i = 1, \dots, n_{nd}, j = 1, \dots, n_\lambda\}$  the set of non-dominated sub-EFs;  
Weights vectors  $\{W_{i,j} : i = 1, \dots, n_{nd}, j = 1, \dots, n_\lambda\}$ ;  
Costs  $\{C_{i,j} : i = 1, \dots, n_{nd}, j = 1, \dots, n_\lambda\}$ .

- 1: re-order returns vector and covariance matrix to give returns in descending order
- 2: set mincost( $1, \dots, n_\lambda$ ) = 1000 and  $n_{nd} = 0$
- 3: count number of cases:  $n_{cases} = \binom{n}{K}$
- 4: **for all** case =  $1, \dots, n_{cases}$  **do**
- 5:   find maximum return and corresponding variance  $R_{max}, V_{max}$
- 6:   find minimum variance and corresponding return  $V_{min}, R_{min}$
- 7:   dominated = TRUE
- 8:   **for all**  $j=1, \dots, n_\lambda - 1$  **do**
- 9:     for each point  $(V_j, R_j)$  on the current EF, compute
- 10:      $p_1 = (V_j - V_{max})(R_{j+1} - R_{max}) - (V_{j+1} - V_{max})(R_j - R_{max})$
- 11:      $p_2 = (V_j - V_{min})(R_{j+1} - (3/4R_{min} + 1/4R_{max}))$   
 $\quad - (V_{j+1} - V_{min})(R_j - (3/4R_{min} + 1/4R_{max}))$
- 12:      $p_3 = (V_j - V_{min})(R_{j+1} - R_{min}) - (V_{j+1} - V_{min})(R_j - R_{min})$
- 13:     **if** case = 1 or  $\min(p_1, p_2, p_3) < tol$  **then**
- 14:       dominated = FALSE
- 15:     **end if**
- 16:   **end for**
- 17:   **if** dominated = FALSE **then**
- 18:      $n_{nd} = n_{nd} + 1$ :
- 19:     **for all**  $j = 1, \dots, n_\lambda$  **do**
- 20:       define  $V_j = W_j^T \sigma W_j$ ;  $R_j = W_j^T R$ , where  $W_j(1, \dots, k)$  is the weights  
vector to be found for each portfolio component
- 21:       find the QP solution for  $W_j$  by minimising the cost function  
 $C(case, \lambda_j) = (\lambda_j V_j - (1 - \lambda_j) R_j)$
- 22:       store  $S_{n_{nd}} = \{(V_{n_{nd},j}, R_{n_{nd},j})\}$
- 23:       store weights vector as  $W_{n_{nd},j}$  and cost as  $C_{n_{nd},j}$  for  $\lambda_j$
- 24:       **if**  $C_{n_{nd},j} < \text{mincost}(\lambda_j)$  **then**
- 25:         mincost( $\lambda_j$ ) =  $C_{n_{nd},j}$
- 26:         update ‘current EF’  $T_j = (V_j, R_j)$ , volatility and return pair for  $\lambda_j$
- 27:       **end if**
- 28:     **end for**
- 29:   **end if**
- 30: **end for**

---

Algorithm 1 provides a complete solution to the 1-variable Markowitz formulation. Note that  $n_\lambda$  denotes the number of points on each non-dominated sub-EF, with  $n_{nd}$  being the number of such sub-EFs found by the algorithm.

As can be seen from Figure 2 and as noted elsewhere (Chang et al., 2000) this formulation leaves significant gaps in the final CCEF. This is due to the necessary concavity of the solution; i.e., the gradient of the final [risk,return] EF in this formulation is (strictly) decreasing. These gaps are not present in the two-objective formulation of the problem. It is possible to fill in many of the gaps on the final CCEF by using a sifting process (Algorithm 2) on the full (pooled) set of non-dominated sub-EFs generated by Algorithm 1, thereby treating the computed CCEF as a union of non-dominated sub-EFs. The resulting CCEF then contains all the non-pointwise-dominated points of the non-curvewise-dominated sub-EFs. As denoted in Algorithm 2, there are  $n_{\max} \leq n_{nd} \times n_\lambda$  points in the final (sifted) CCEF.

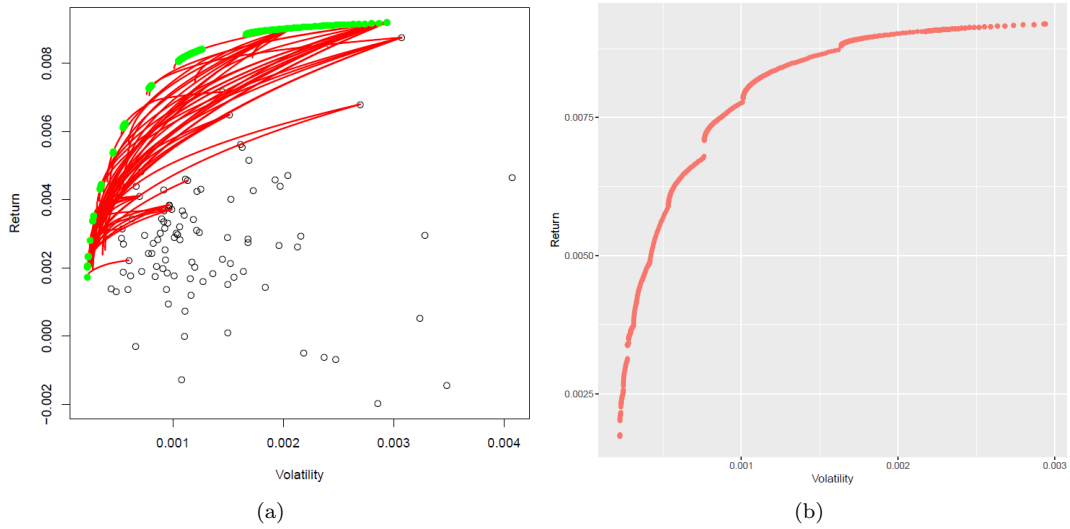


Figure 2.: Production of a CCEF with dataset (D4) and cardinality  $K = 3$ . Subfigure (a) shows the output from Algorithm 1 (non-dominated points shown in green, and the 54 non-dominated sub-EFs in red), with (b) showing the result of the sifting from Algorithm 2. Note the ‘gaps’ and ‘bunching’ at several return values.

The resulting weights may be extracted from the final CCEF. In principle, the ‘*tol*’ parameter (which determines whether or not a given sub-EF is ‘below and to the left’ of all points on the ‘current’ efficient frontier) should be set to zero. However, setting *tol* to zero generates an excessive number of sub-EFs identified as being curvewise non-dominated but which the sifting process actually identifies as being pointwise dominated, and thus not contributing to the final CCEF. Setting *tol* to a small positive value avoids this problem. A value of  $tol = 10^{-25}$  was found to be suitable for the current study. Note that the value of ‘*tol*’ is likely to depend both upon the precision of the arithmetic used in floating-point calculations and upon the magnitudes of the returns observed in a given dataset. A reasonable strategy in the general case would be to start from a relatively large value of ‘*tol*’ that identifies too few non-dominated sub-EFs (e.g.,  $10^{-10}$  for the datasets considered here), then decrement in powers of ten until the number of sub-EFs is stable.

The code was extended in R from the original Maple code detailed in the preliminary work of Craven and Graham (2017), and is available at <https://github.com/>



---

**Algorithm 2** Pooling/sifting

---

**input:** pooled  $n_{nd} \times n_\lambda$  points from non-dominated sub-EFs  
**output:** updated  $n_{\max}$ : number of non-dominated EF points

```
1:  $i \leftarrow 1$ 
2: while  $i < n_{\max}$  do
3:   find  $\max_j R(j)$ ,  $j = i, \dots, n_{\max}$ 
4:   swap  $(V(\max_j), R(\max_j))$  with  $(V(i), R(i))$ 
5:   for all  $j = i + 1, \dots, n_{\max}$  do
6:     remove  $(V(j), R(j))$  if  $V(j) \geq V(i)$ 
7:     re-index remaining points  $j > i + 1$ 
8:      $n_{\max} \leftarrow n_{\max} - 1$ 
9:   end for
10:   $i \leftarrow i + 1$ 
11: end while
```

---

MJCraven/SiftedQP. All experiments are run on an Intel Xeon E5-2683v4 processor at 2.10 GHz with a relatively low amount of memory needed. The approach is deterministic. The next section gives results and observations from the method.

## 4. Results and Observations

### 4.1. Datasets Used

To demonstrate the approach, the standard benchmark instances (D1)–(D5) of OR-Library (Beasley, 1990) will be used. These instances cover returns data over the period March 1992 to September 1997 for 31, 85, 89, 98 and 225 assets (encompassing the Hang Seng, DAX 100, FTSE 100, S&P 100 and Nikkei 225 indices) respectively. In addition, two further datasets will be used. Dataset 389 is from the daily prices of all assets in the S&P500 index for the period 18/11/1999–09/08/2013 and dataset 493 is from the S&P500 index over the shorter period 10/08/2012–09/08/2013 (Craven & Graham, 2019). In each case, assets were excluded if they were not constituents of the index for the full period under consideration, hence the inclusion of a larger number of assets for the shorter period. Table 1 gives the effective ranges of  $\lambda$  used for experiments; the lower limit,  $\lambda_0$ , is non-zero in many cases since for  $0 \leq \lambda_i < \lambda_0$  the co-ordinates  $(r_i, \bar{R}_i)$  may be identical. That is, the lower limit on each interval gives the value of  $\lambda$  corresponding to the point with maximal return (top right of the EF), and the upper limit  $\lambda_1 = 1$  the point with minimal risk (bottom left of the EF).

Table 1.:  $\lambda$  ranges used for each dataset (from Craven and Graham (2019)).

Instance	(D1)	(D2)	(D3)	(D4)	(D5)	389	493
$[\lambda_0, \lambda_1]$	[0.342,1]	[0.149,1]	[0.426,1]	[0.107,1]	[0.115,1]	[0,1]	[0.489,1]

In the following results, the time to compute the estimated sub-EFs for the identified cases and also the time to sift the sub-EF collection to produce the CCEF are given. As the algorithm is deterministic, the sub-EFs produced are constant over that set of ten runs. All times are mean times over ten runs with each set of parameters  $n$ ,  $K$  and  $n_\lambda$ . In various cases, the solutions obtained are compared with those of Cesarone et al.

(2013). In most cases (e.g., Figure 3) the comparison is excellent. In some instances, however, there are slight discrepancies (see Figure 9 and related discussion in Section 4.4 for further details).

#### 4.2. Test of Scalability

The first experiments run were with instance (D1),  $K = 2, 3, 4, 5, 6, 8$  and varying  $n_\lambda = 10, 50, 100, 200, 500$  to observe the scaling behaviour of the algorithm (Table 2). Various data were taken from the runs: the number of non-dominated sub-EFs (also referred to as the identified cases), the number of final CCEF points after sifting, the computation time and the sifting time. Computation time is taken without the sifting process. The number of sub-portfolios,  $\binom{n}{K}$ , is also recorded for comparison purposes.

Table 2.: Scalability test. Observe that the number of sub-EFs does not strictly increase; this may be due to the locations  $\lambda$  values are placed along the CCEF.

$K$	$n_\lambda$	# CCEF pts	# ND Sub-EFs	# Sub-PFs	Comp. Time (s)	Sift Time (s)
2	10	41	6	465	0.08	0.24
	50	197	9		0.16	0.31
	100	389	9		0.27	0.51
	200	756	9		0.49	1.22
	500	1869	9		1.14	6.14
3	10	33	7	4495	0.59	0.23
	50	137	12		1.25	0.31
	100	248	12		2.07	0.55
	200	460	13		3.76	1.46
	500	1107	13		8.70	7.52
4	10	36	8	31465	4.01	0.22
	50	155	15		8.51	0.37
	100	297	17		14.16	0.85
	200	547	17		25.35	2.51
	500	1318	17		58.75	13.93
5	10	46	10	169911	22.05	0.21
	50	206	19		46.90	0.46
	100	401	21		76.49	1.26
	200	711	22		137.45	4.32
	500	1708	22		315.28	24.70
6	100	468	24	736281	336.37	1.58
	200	865	24		594.21	5.23
	500	2048	24		1370.64	30.35
8	100	589	23	7888725	3820.53	1.69
	200	1103	23		6488.53	5.63
	500	2557	23		14840.66	32.09

Some observations on the results of Table 2 are as follows. Increasing the value of  $K$  uniformly results in computational time increasing, expected due to the increased number of sub-portfolios. However, for  $K < 5$ , computation time takes under one minute even for large numbers of points. Fixing  $n_\lambda = 200$ , analysis of the  $K = 8$  case

suggests that around 105 million combinations may be analysed in one day with no use of parallel computing.

Assuming  $K$  is fixed, increasing the value  $n_\lambda$  seems to imply that the number of non-dominated sub-EFs converges. For example, the algorithm seems to converge to nine non-dominated sub-EFs for  $K = 2$  and twenty-four for  $K = 6$ . In addition, the number of points on the final CCEF increases in line with  $n_\lambda$ , producing high resolutions of over 2000 points for  $n_\lambda = 500$ . Unsurprisingly, sifting takes longer for larger values of  $n_\lambda$ , due to having to sift  $n_\lambda$  points on each non-dominated sub-EF. As an example, Figure 3 shows the contrast in solution quality between fifty and five hundred  $\lambda$ -values for  $K = 3$ .

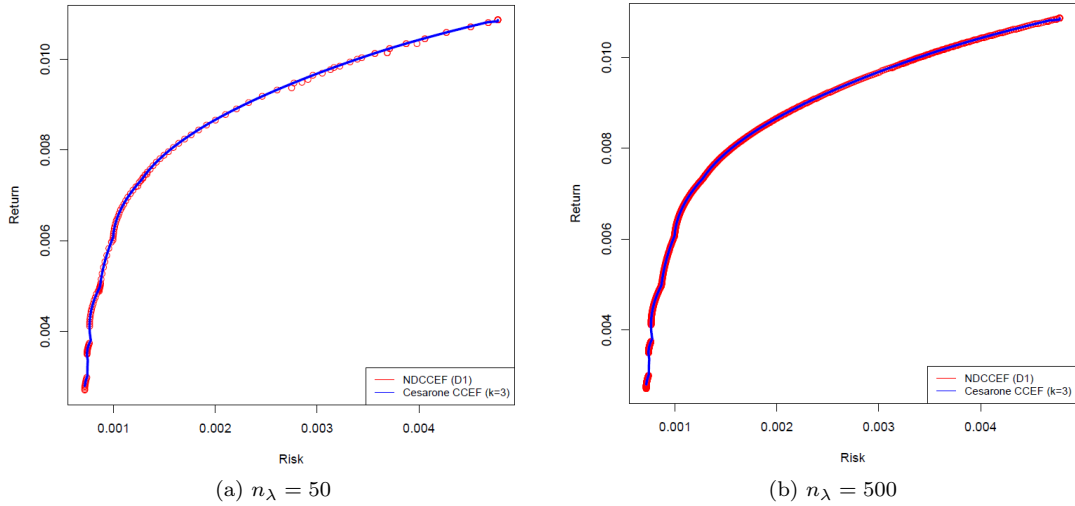


Figure 3.: Approximation of the (D1) CCEF of Cesarone et al. (2013) (blue lines) by the computed non-dominated CCEF (red dots). As in Table 2, 137 dots are present on the left subfigure and 1107 on the right.

Hence, by the above, an excellent approximation to the CCEF (via the sifted CCEF) may be made by consideration of only a small fraction of the number of sub-EFs, indicating the efficiency of the approach. The next subsection details similar experiments on the remaining datasets.

#### 4.3. Experiments on Other Datasets

In this subsection, experiments are performed on the remaining datasets with values of  $K$  such that the number of sub-portfolios is less than ten million. Two hundred points ( $n_\lambda$ ) on each sub-EF were used as this provided a resolution of at least 500 points on the final CCEF after sifting. The results are given in Table 3 and, together with those from Table 2, are available at Graham and Craven (2019).

It is well-known that the CCEF becomes closer to the unconstrained EF as the value of  $K$  increases (Chang et al., 2000). That is, even though the number of non-dominated sub-EFs is a small fraction of the total number of sub-portfolios, the CCEF and the unconstrained EF will coincide for sufficiently large  $K$  (or become close enough for the difference to become largely irrelevant). At this point the method is outperformed by traditional QP. There is a clear correspondence between the computation time and the number of sub-portfolios, and between the sift time and the number of non-dominated

Table 3.: Results for datasets (D2)–(D5), 389 and 493, all with  $n_\lambda = 200$ .

Instance	$K$	# CCEF pts	# ND Sub-EFs	# Sub-PFs	Comp. Time (s)	Sift Time (s)
(D2)	2	552	13	3570	3.14	1.58
	3	838	17	98770	81.69	2.77
	4	1286	19	2024785	1690.17	4.73
(D3)	2	592	10	3916	3.45	1.26
	3	670	24	113564	94.81	4.24
	4	837	44	2441626	2048.02	12.15
(D4)	2	1218	27	4753	4.31	6.95
	3	1073	54	152096	127.92	16.91
	4	1231	63	3612280	3061.69	22.13
(D5)	2	767	14	25200	23.19	2.04
	3	769	31	1873200	1709.75	6.16
389	2	1393	48	75466	75.70	16.88
	3	1707	51	9735114	9894.78	21.78
493	2	614	61	121278	124.78	17.09

sub-EFs. Except for  $k = 2$  for the smaller datasets, the sift time is relatively small as a proportion of the computation time.

Figure 4 shows results from instance (D3) with cardinality  $K = 3$ . Subfigure (a) gives an analogue of Fig 2(a), comprising the 24 non-dominated sub-EFs found in red, with (b) giving a comparison of the non-dominated CCEF (670 points) obtained with the CCEF of Cesarone et al. (2013). Note the presence of a portion of the solution with negative returns, which is not found in the results of the aforementioned authors. Finally, (c) gives a heatmap for the asset proportions of a sample of final CCEF points. In this heatmap, every tenth point is sampled and non-negligible proportions are represented by colours. The horizontal axis is ordered by the ranks of the utilised assets from highest return (left) to lowest return (right); the non-utilised assets are omitted. The vertical axis corresponds to the index of each of the  $n_{max}$  points of the final CCEF. Index 1 represents the ‘top-right’ point of maximum return of the CCEF corresponding to a portfolio containing only the single highest-returning asset. Index  $n_{max}$  represents the ‘bottom-left’ point of minimum risk, corresponding to a portfolio with contributions from several assets. The heatmap shows a clear progression of contributions moving away from the highest-returning asset towards a blend of lower-returning assets as return and risk decrease along the CCEF.

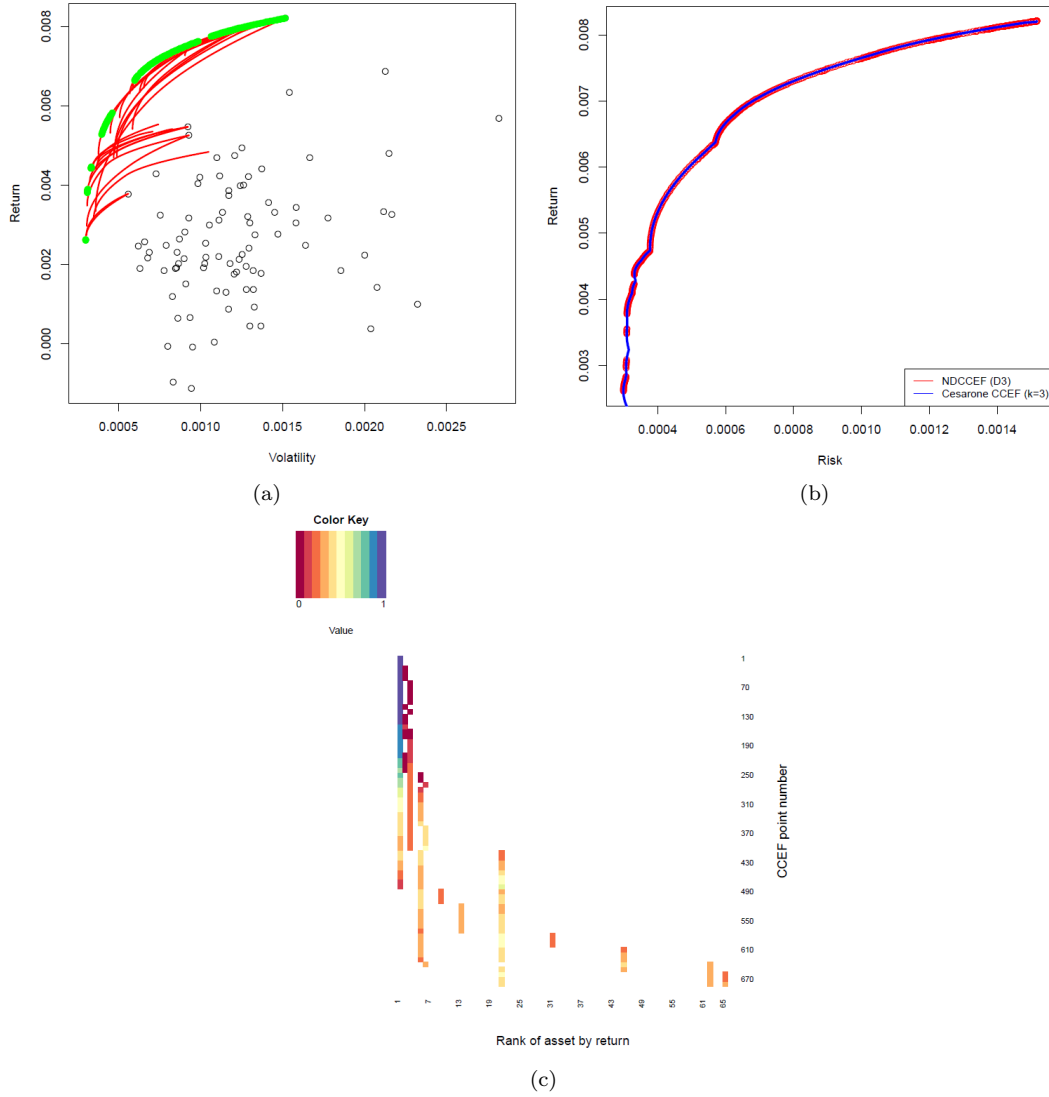


Figure 4.: Plots of results from instance (D3) with  $K = 3$ . Plot (a) shows the 24 non-dominated sub-EFs in red, and (b) the full convergence of the CCEF by the approach. The sampled heatmap is shown in (c), white denoting weight zero.

Figures 5-8 give plots produced by the approach for other instances. In some cases, the approach finds points not on the CCEF of Cesarone et al. (2013), noting that most of these are negative in return (e.g., Figs 5 and 6).

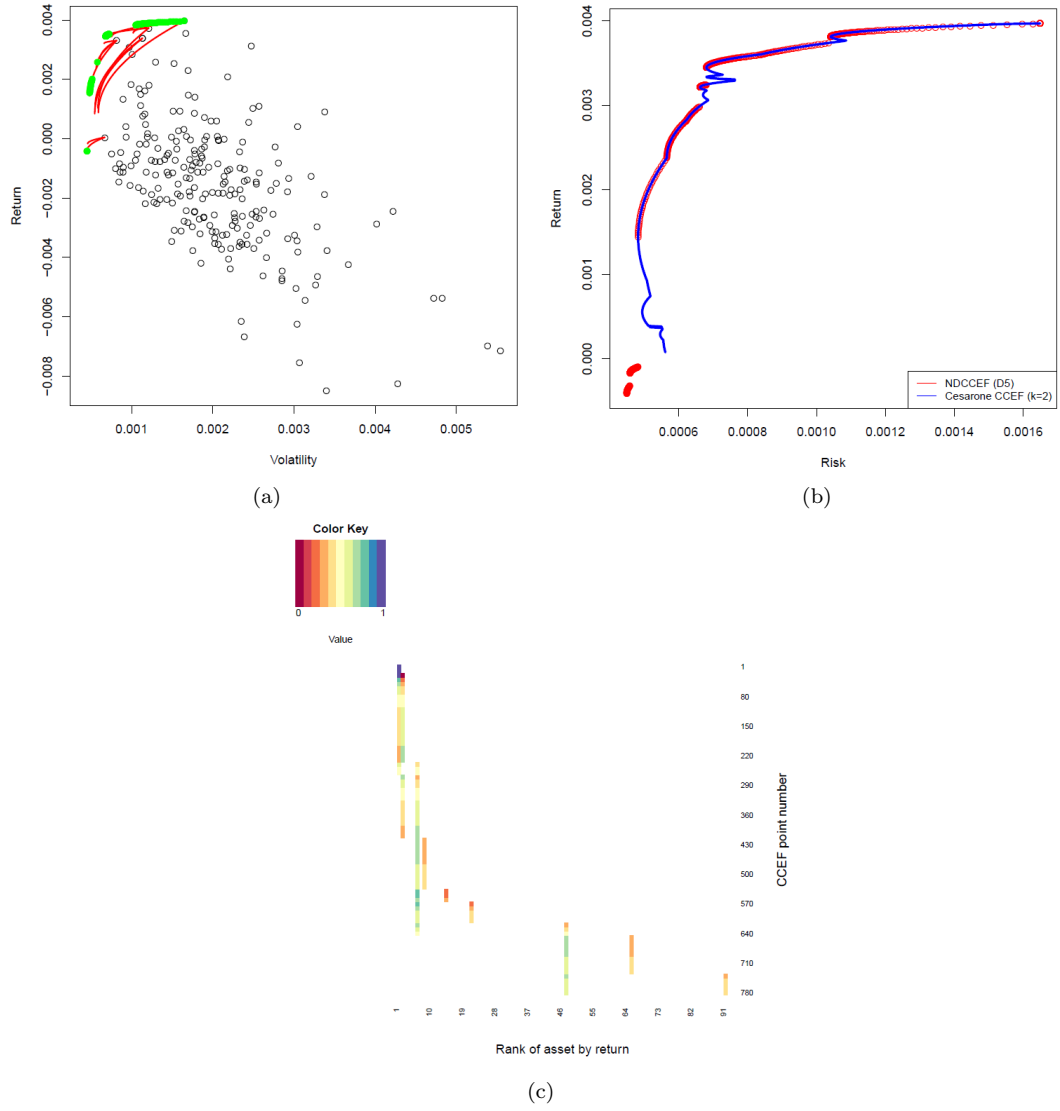


Figure 5.: Solution comparison of the (D5) instance with  $K = 2$ . Note the presence of two negative parts of the solution with the present approach. The non-coverage of the branch of the comparison solution (blue) around return levels of 0.0014 and below is present since the branch is dominated by the red dot immediately above.

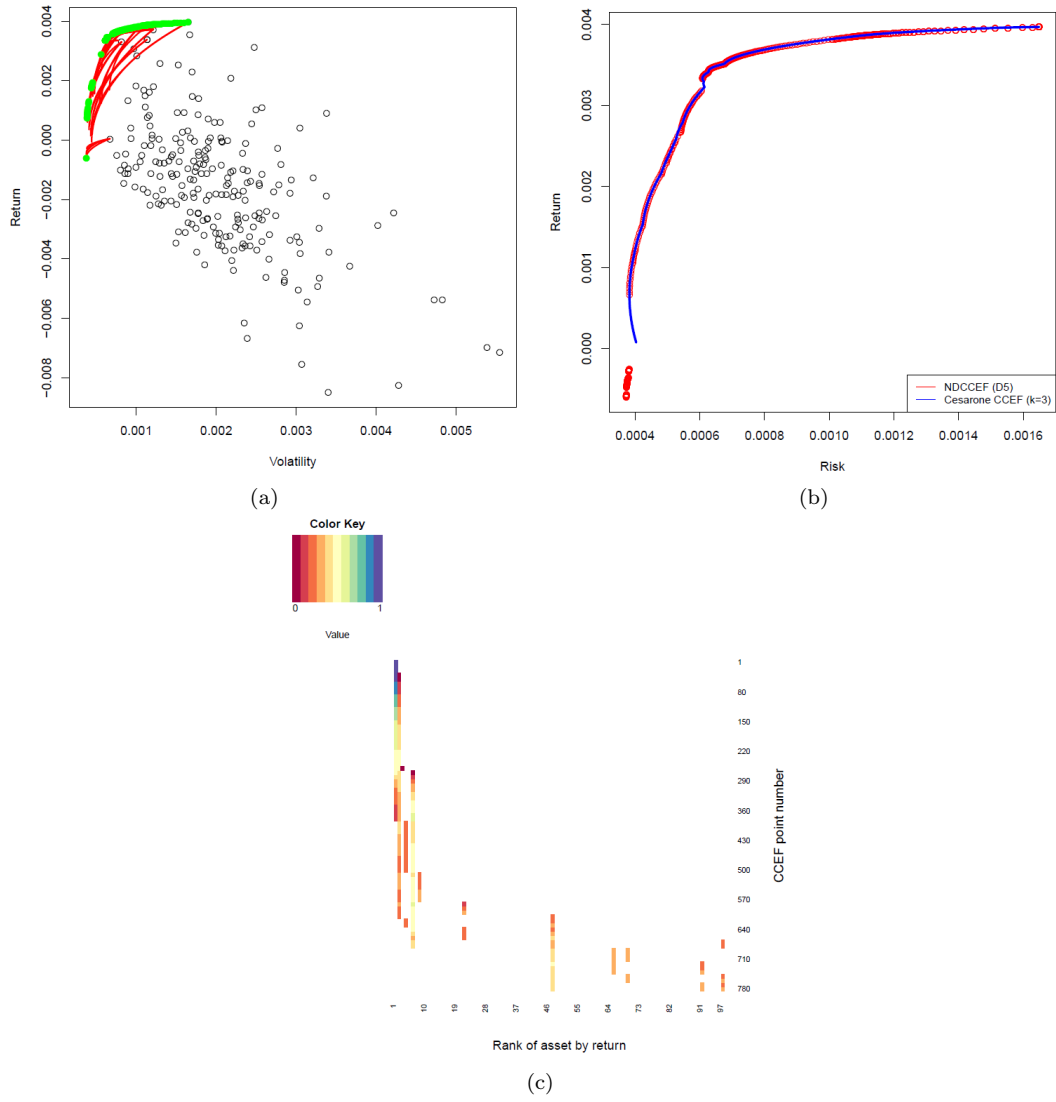


Figure 6.: Results from (D5) with  $K = 3$ , plot (a) being the non-dominated sub-EF visualisation, (b) being the comparison of the CCEFs produced by the approach (red) and the method of Cesarone et al. (2013) (blue), and (c) the sampled heatmap.

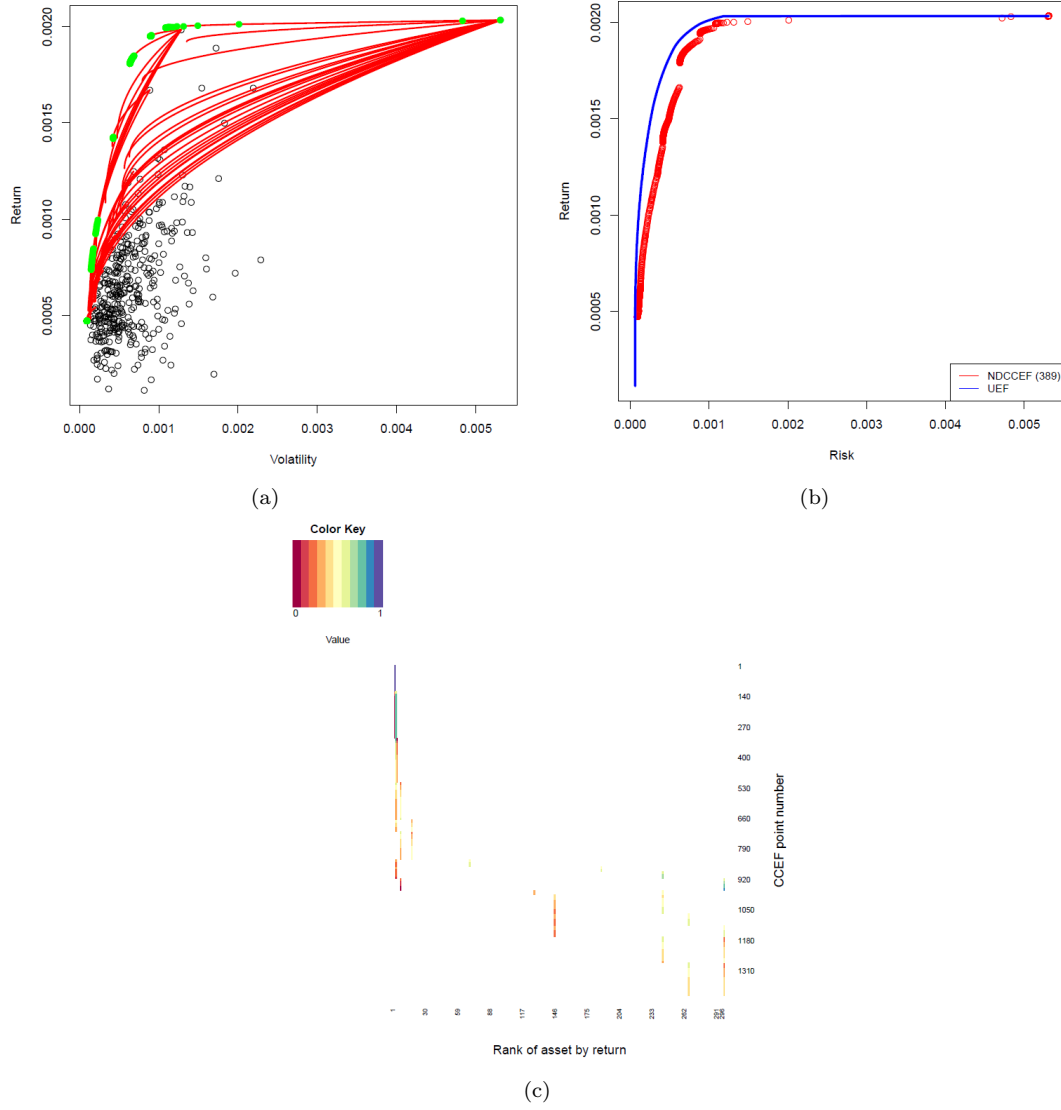


Figure 7.: Solutions of the S&P 389 dataset with  $K = 2$ . In (b), the CCEF is compared with the unconstrained EF since the true CCEF is unknown.



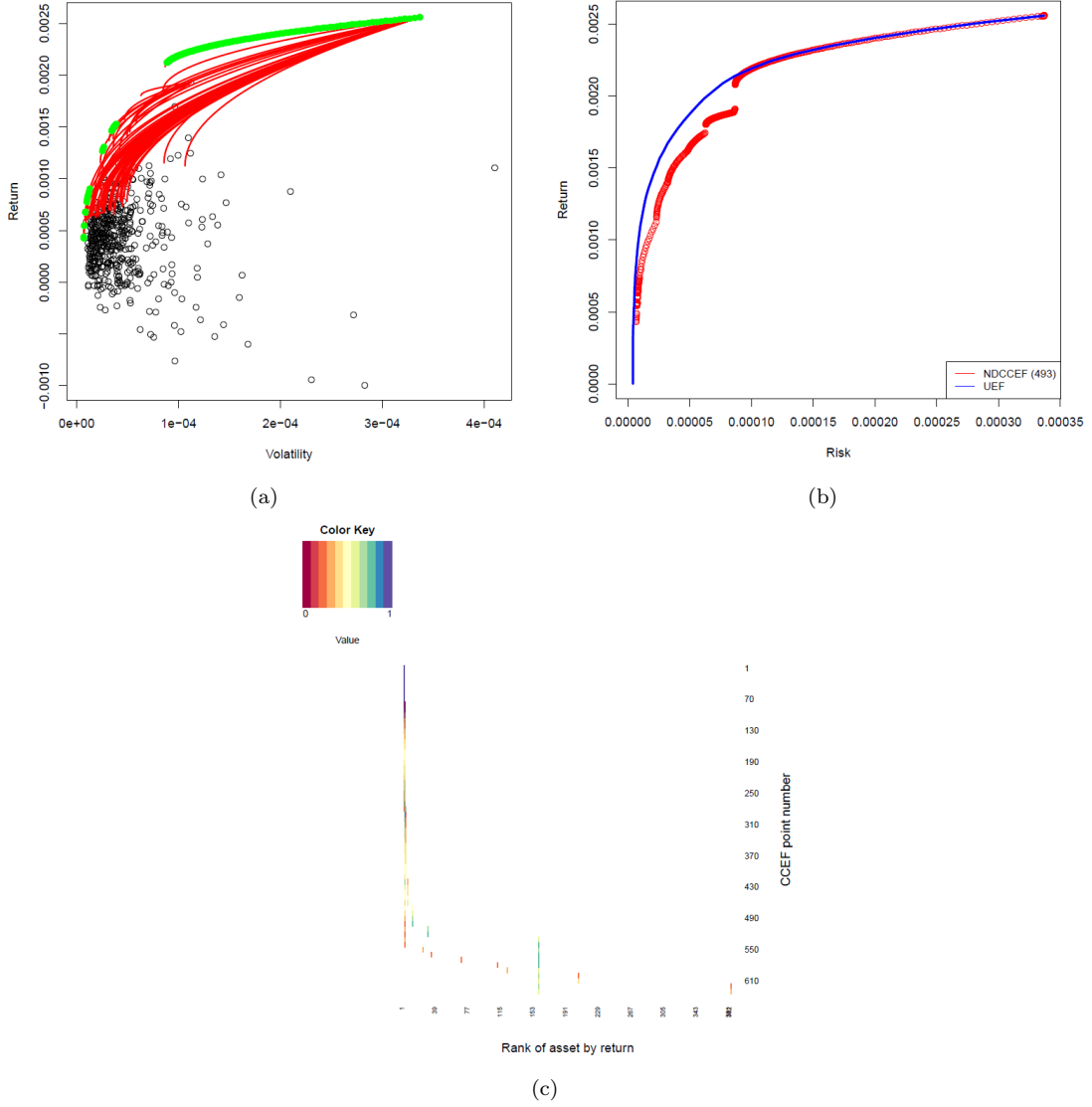


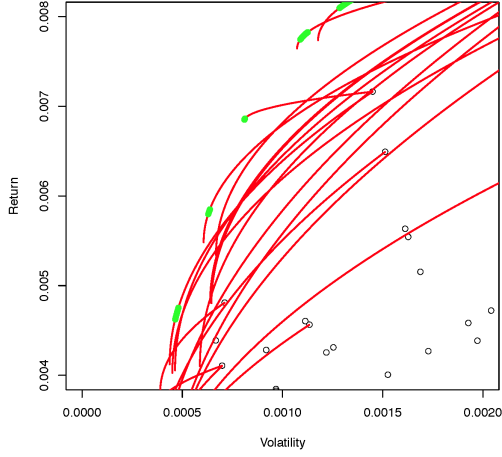
Figure 8.: Solutions of the S&P 493 dataset with  $K = 2$ . In (b), the CCEF is compared with the unconstrained EF since the true CCEF is unknown.

For all cases in this work, the vast majority of sub-portfolios are clearly dominated. This is clear from the plots shown and means that only a very small fraction of the total number of sub-portfolios need to be computed. Further, from the computation times given in Tables 2 and 3, the sub-portfolios are computable within a short time.

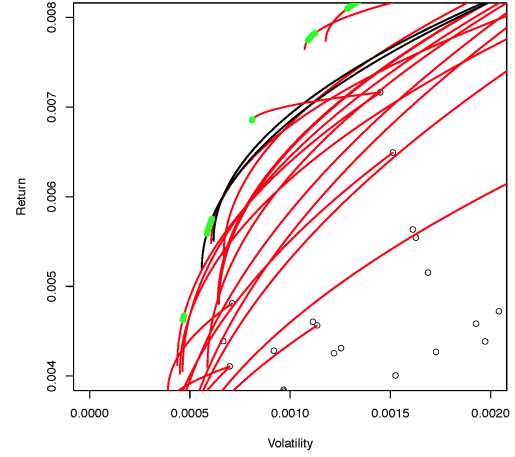
#### 4.4. Identification of Non-Dominated Sub-EFs

Final efficient frontiers computed with the approach are determined from non-dominated sub-EFs. For a given sub-EF, the endpoints corresponding to  $\lambda = 0$  and  $\lambda = 1$  are  $(V_{\min}, R_{\min})$  and  $(V_{\max}, R_{\max})$  respectively, where  $(V_{\min}, R_{\min})$  is the minimum risk and its associated return and  $(V_{\max}, R_{\max})$  is the maximum return and the associated risk for the sub-EF in question. As noted in Section 3.1 above, these points are either known from the data  $[(V_{\max}, R_{\max})]$  or computed using QP

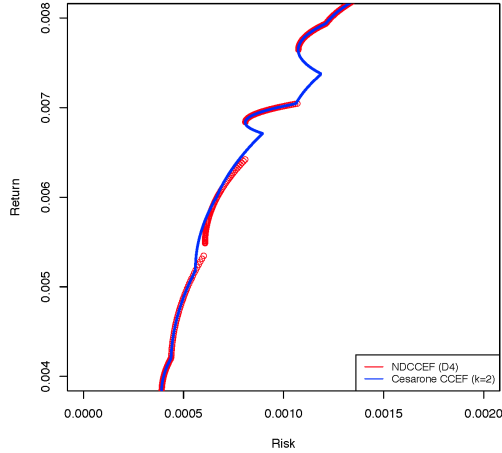
$[(V_{\min}, R_{\min})]$ . We then test for pointwise-dominance these two points, together with a third point located at  $(\frac{1}{2}(V_{\min} + V_{\max}), \frac{1}{4}R_{\max} + \frac{3}{4}R_{\min})$ . The choice of the third point rests upon the assumption that, for the sub-EF, the return  $R$  is a quadratic function of the risk  $V$ . This is exact for  $K = 2$  but is otherwise an approximation. Results are slightly sensitive to the choice of this ‘third point’. For example, Figure 9 (c) shows that the final EF computed using the approach for the case  $K = 2$  for portfolio (D4) coincides with that from Cesarone et al. (2013) except for return values between 0.0055 and 0.0065, where the approach slightly under-performs.



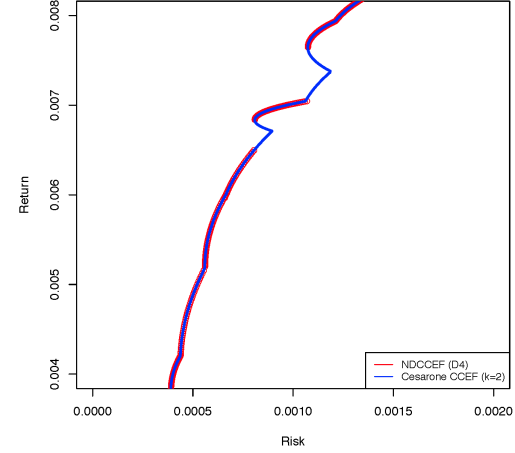
(a) Non-dominated sub-EFs  $\alpha = 0.75$ .



(b) Non-dominated sub-EFs  $\alpha = 0.85$ . Additional sub-EFs are coloured black.



(c) Final CCEF  $\alpha = 0.75$ .



(d) Final CCEF  $\alpha = 0.85$ .

Figure 9.: Enlarged comparisons of non-dominated sub-EFs (a), (b) and final CCEF (c), (d) results for (D4) with  $K = 2$  using  $\alpha = 0.75$  (a), (c) and  $\alpha = 0.85$  (b), (d).

However, locating the third point at  $(\frac{1}{2}(V_{\min} + V_{\max}), (1 - \alpha)R_{\max} + \alpha R_{\min})$  for  $\alpha = 0.85$  identifies two further non-dominated sub-EFs compared with the standard choice  $(\frac{1}{2}(V_{\min} + V_{\max}), \frac{1}{4}R_{\max} + \frac{3}{4}R_{\min})$  (compare 9 (a), (b), with the two additional sub-EFs coloured black in 9 (b)). This results in an exact match with the data from Cesarone et al. (2013) - (compare 9 (c), (d)). Higher values of  $\alpha$  could be used, but this may result in the generation of a greater number of non-dominated sub-EFs and thus

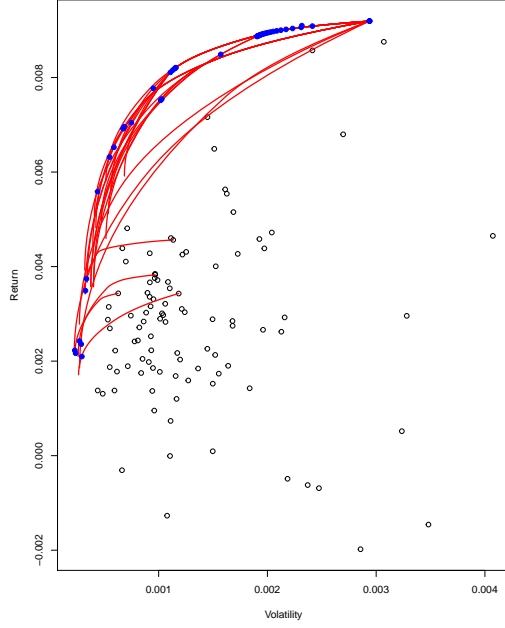
increased computation times. It has been found that there is no universally-acceptable value for  $\alpha$  other than the standard 0.75 obtained using the quadratic assumption and the standard value generally offers a reasonable compromise between acceptable accuracy and acceptable computation times. An alternative strategy would be to vary  $\alpha$  incrementally from a starting value of 0.75. At some stage, the number of sub-EFs may become excessively high and thus the final value of  $\alpha$  could be set as the maximum value at which the number of sub-EFs is not excessive.

This section concentrated upon the generation of CCEFs for a number of datasets. As stated previously, it is known that the CCEF may be efficiently approximated by metaheuristics such as EAs for reasonable values of  $K$ . It turns out that visualisation of the non-dominated sub-EFs found by the approach in this work may shed light on the evolution process of EAs. This is exposed next.

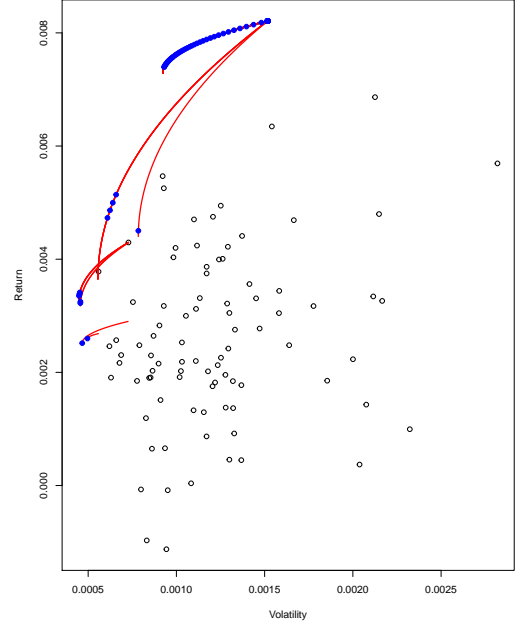
## 5. Relation to EA Solution Processes

This work has given an overview of the approach being an alternative to EAs on the cardinality constrained problem, especially for small cardinalities (values of  $K$ ). However, the analysis can be related to the results generated by EAs for such problems. EAs are driven by objective functions, which encapsulate key pieces of information from the problem and which must be optimised in the process of finding as optimal a solution as possible. It turns out that the approach and EAs have commonalities. Corresponding to the one-objective formulation of the problem, only the case of a single-objective EA is considered in this work. It is also assumed that an EA is attempting to minimise its objective function (clearly, the EA maximises if the objective function is negated).

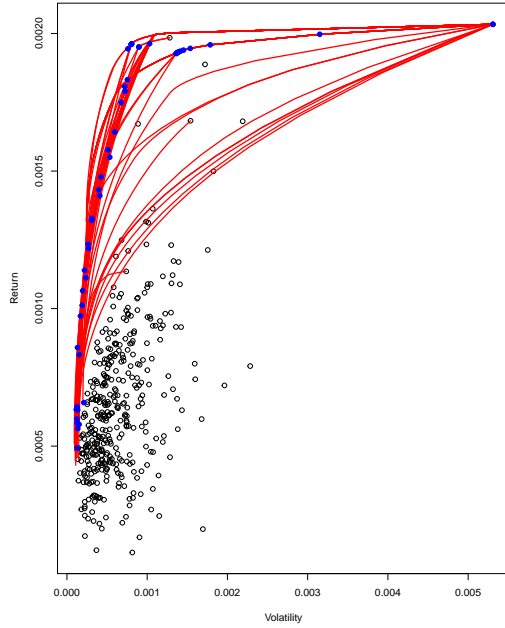
A well-known disadvantage of using an EA (and similar methods) on a problem of this complexity is the presence of local objective minima in the evolution solution process (Michalewicz & Fogel, 2013). The following shows that, for the efficient frontier problem, this phenomenon can be understood in terms of the EA solution being trapped on non-optimal (i.e. *dominated*) sub-EFs. The EA solution process appears to select a particular sub-EF, or a portion thereof, after which use of a mutation operator corresponds to the solution tracing along that sub-EF as the  $K$  portfolio weights are perturbed. The EA's crossover operation allows the process to jump between sub-EFs. For example, Figure 10 (a, b, c, d) shows such a picture for portfolios of sizes  $n = 85, K = 2$  (a),  $n = 98, K = 2$  (b),  $n = 389, K = 15$  (c) and  $n = 493, K = 30$  (d). The unfilled dots are the original values of risk and return from the individual assets and the blue filled dots give the points on the EF generated by the EA of Craven and Graham (2019). The red curves are sub-EFs associated with the particular sub-portfolios identified from the weights matrix generated by the EA. Each row of this matrix contains non-zero weights from up to  $K$  assets and the sub-EFs are generated (using QP; i.e., independently of the EA) from these assets. It is clear that any given point of the EF computed from the EA lies upon the sub-EF associated with the same subset of  $K$  assets as used by the EA.



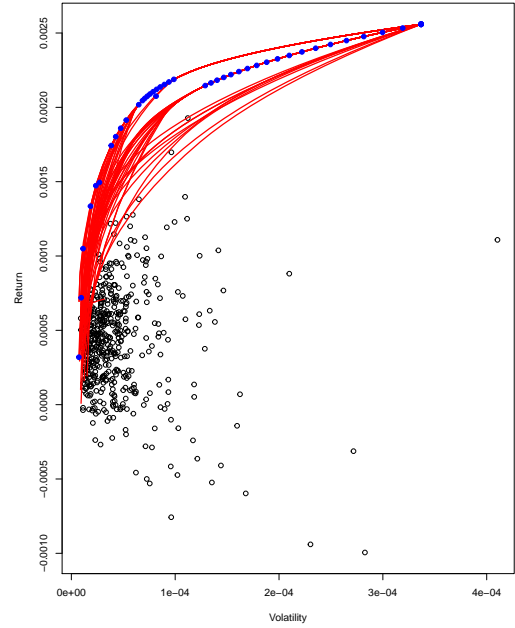
(a) (D2) with  $K = 2$ .



(b) (D3) with  $K = 2$ .



(c) 389 with  $K = 15$ .



(d) 493 with  $K = 30$ .

Figure 10.: Relation between points computed by EA (blue) and QP-generated sub-EFs using the same assets (red). Note that the S&P389 dataset seems to be distinct to the other treated datasets since there is a large part of the plot that has a very small gradient.

This indicates that possible enhancements of the EA could be found by utilising a crossover process that uses this knowledge of the topography of the solution space to allow more rapid exploration of this space, enabling the EA to be more likely to land upon non-dominated sub-EFs and hence optimal solutions. Comparing the approach described in this paper with other existing approaches, thoughts turn to other exact methods and metaheuristics. This is discussed in the next section.

## 6. Comparison of the Current Approach with Other Works

### 6.1. Metaheuristics

To the best knowledge of the authors, there are no metaheuristics that exhibit solutions to the benchmark cardinality constrained problems (D1)-(D5) for  $K < 5$ . Typically, many works consider these instances with  $K = 10$  (Anagnostopoulos & Mamanis, 2011; Chang et al., 2000; Lwin et al., 2014), although some works consider cardinalities  $K = 5$  and  $K = 15$  (mentioned in the work of Lwin et al. (2014), with details available in the PhD thesis of the first author, and Anagnostopoulos and Mamanis (2011) use  $K = 5$  and  $K = 10$  with a non-benchmark dataset). This paucity of results for  $K < 5$  suggests that good quality solutions for such values of  $K$  are hard to achieve for metaheuristics. Also, the current approach is not used with  $K = 10$  since the dominating term,  $\binom{n}{K}$ , makes the search inherent in this work impractical. Hence a comparison of metaheuristics and the current approach, where typically  $K < 5$ , is believed to be also impractical.

On a methodological level, metaheuristics are algorithms which specialise in finding approximate solutions quickly and are essentially stochastic in nature. The current approach is, by contrast, deterministic. Hence the current approach is very different from metaheuristics, methodologically as well as quantitatively.

### 6.2. Exact Methods

In this subsection, the current approach is compared with other exact methods. The first comparison algorithm is that of Cesarone et al. (2013), with which a graphical comparison was conducted in Section 4. In terms of time complexity and efficiency, the work of Cesarone et al. (2013) studies the problem with 1% minimum asset proportion and finds 500 CCEF points, fewer in many cases than the current approach. However, the computation environment on which that work was conducted, reported as an Intel Core2 Duo T7500 CPU at 2.2 GHz, is at least a computational generation apart from the current environment and hence time comparisons are not made. Cesarone et al. (2013) point out that their method exhibits “exponential complexity in the worst case”, but through modification “can achieve polynomial time complexity”. As explained below, the current approach runs in polynomial time in  $n$  for small  $K$ . It was reported that, for  $K = 5$  and  $K = 10$ , computation times using their method were generally smaller than those found using the CPLEX solver. CPLEX computation times for  $K = 5$  were larger than for  $K = 10$ .

The following two works examine the problem from a different point of view, relating to its solution at one return value rather than a selection of values across the CCEF. The work of Gao and Li (2013) analyses a variation of a branch-and-bound algorithm on the problem under a Lagrangian relaxation. The return value is restricted to “the median of the expected returns of the given pool of stocks”. Results are presented from a 2.8 GHz CPU, giving numbers of examined nodes and execution times that

outperform CPLEX for a selection of problem sizes. Contrary to the above work, the  $K = 5$  cases are computed more rapidly than those for  $K = 10$ . The OR-Library instances are not used, but some broadly equivalent problem sizes are presented ( $n = 80$  and  $n = 100$ ).

The work of Bertsimas and Cory-Wright (2019) discusses a cutting-plane method ran on an Intel i9 2.9 GHz CPU and 16 GB RAM using the OR-Library instances. The return value is set to be 0.3 of the way up the CCEF, the times reported comparing well with a reformulation solved in CPLEX. As in the work of Cesarone et al. (2013),  $K = 10$  computation times are generally larger than those for  $K = 5$ .

Computation times for both these works are given in Table 4, where (as stated) the dataset size of Gao and Li (2013) is equated to the nearest OR-Library instance in size. Although these works used different CPUs, both solve the problem for a single point on the CCEF and minimum cardinality  $K = 5$ , whereas the current work is designed to compute the full CCEF. To offer a per-point time comparison the approach was ran on the three instances (D2)–(D4) with  $K = 5$ . The time for (D1) with  $K = 5$  is produced from Table 2. Instance (D5) was not ran with  $K = 5$  since there are over 4.5 billion sub-portfolios. Observe that the mean per-point computation times compare well with those of Bertsimas and Cory-Wright (2019) as well as those of Gao and Li (2013), whilst noting that direct comparisons are not completely reliable due to differences in computational environment.

Table 4.: Stated computation times for  $K = 5$  from the work of Gao and Li (2013) [GL] and Algorithm 3 of Bertsimas and Cory-Wright (2019) [BC-W], compared with the current approach point-by-point. The number of non-dominated sub-EFs and the number of CCEF points generated by the current approach are also presented.

Instance	[GL]	[BC-W]	# ND Sub-EFs	# CCEF points	Comp. time (pt.)
(D1)		0.22	22	711	0.19
(D2)	178.7	48.29	24	3157	19.76
(D3)		175.20	56	2262	35.21
(D4)	492.2	2690.00	76	3919	33.23
(D5)		0.85	N/A	N/A	N/A

It is clear that the cutting-plane and branch-and-bound approaches are more efficient for larger  $K$  than the current method, which is specialised to small  $K$ . Hence, for the reasons stated, a direct comparison of the current method with those works referred to above proves difficult but the current method appears competitive, as well as being conceptually simpler.

### 6.3. A Complexity Estimate for the Current Approach

A full time complexity analysis of the algorithm is out of scope of this work. However, a brief approximation is given as follows. There are three main factors contributing to the computation time of the approach. Firstly, there are the  $\binom{n}{K}$  combinations of assets the algorithm processes. By a standard result,  $\binom{n}{K} \leq \frac{n^K}{K!}$  for  $1 \leq K \leq n$ . Subsequently, the complexity of the QP procedure is bounded above by the complexity of multiplying  $K \times K$  matrices (which is  $O(K^3)$  at worst). The final factor is the number of points

in each sub-EF tested. Thus a crude upper bound on time complexity of

$$O\left(\frac{n^K}{K!}K^3n_\lambda\right) \quad (4)$$

is suggested. It is clear that  $\frac{n^K}{K!} \leq n^K$  and so the approach runs in polynomial time in  $n$  for small  $K$ . The data in Tables 2–4 are broadly consistent with this estimate.

## 7. Conclusion

This work presented an efficient method for construction of small-cardinality constrained efficient frontiers of portfolio problems by the solution of a sequence of smaller QP problems (to produce ‘sub-EFs’), followed by the application of a sifting procedure to the pooled data to produce the final CCEF. These CCEFs are highly accurate, comparing very well to those of Cesarone et al. (2013) for the OR-Library instances of Beasley (1990), and so form a method to efficiently calculate benchmark optimal solutions for EAs and other optimisation algorithms. It has also been shown that the approach performed very well on datasets extracted from the S&P 500 index (the 389- and 493-datasets). The above implies that the approach is an excellent alternative to using the above metaheuristic algorithms to solve portfolio problems for small cardinality constraints. The detailed approach may be performed for any cardinality  $K$  until the user runs out of allocated time budget. Note, however, that (i) cost rises quickly as  $K$  increases, and (ii) for larger values of  $K$  a metaheuristic could approximate the problem efficiently (or QP could be used if  $K$  is sufficiently large and the problem is effectively unconstrained).

Via Section 5, it was shown for a range of datasets tackled that the sub-EFs identified in the solution process served as local minima cost curves of the EA. It has been shown that by using only a small number of  $\lambda$ -values a large number of points on the accurate CCEFs may be found (ranging from 552, compared with Cesarone et al. (2013) solutions of 500 points resolution, to over 1700 points - Table 3 for  $n_\lambda = 200$ ). Further, a visualisation of not only CCEF-formation but also EA solution formation processes has been demonstrated. This enables a ready and fast realisation of where final CCEFs come from and, also, in association with the benchmark solutions produced, identification of some EA local optima (although by no means all). This gives insight into where such approaches exhibit difficulties in finding optimal solutions.

A comparison between the current work, metaheuristics and exact methods was given in Section 6. Methodological differences were detailed, as well as computational complexity for exact methods. Despite these methodological differences, it was found that the computation time of the current work compares well with that of state-of-the-art works such as Bertsimas and Cory-Wright (2019). In terms of solution quality, highly accurate optimal solutions are produced; these typically are more accurate than those of metaheuristics (see Figures 10(a), (b) for an example of an EA solution of two small-cardinality problems - several blue points are on sub-optimal non-dominated sub-EFs). Finally, the approach offers a more intuitively simple (and thus more generalisable) solution than that of Bertsimas and Cory-Wright (2019), Cesarone et al. (2013) and Gao and Li (2013). However, the approach does have limitations: the main limitation is the size of the search space that is enumerated (albeit quickly). If a domination criterion which reduces the computational effort required in this search can be found then this limitation becomes less serious, allowing the production of CCEFs

for larger  $K$  than in the present study. However, the strengths of the approach do outweigh the limitations in that there are several avenues of further work that will be pursued.

The first avenue of further work is to generalise the methodology to include different risk measures, especially non-quadratic risk measures. For alternative risk measures to be workable, it is observed that a definition (possibly entirely distinct from the current definition) of curvewise dominance with respect to the risk measure would be required. In addition, a fast nonlinear optimisation method, depending upon risk measure employed, to replace QP minimisation would be required. Related to this are different versions of portfolio problems, with modifications to the approach as required. This includes the possibility of two- or more objective versions of portfolio problems being tackled, to enable finding the ‘kinks’ and highly non-linear parts of the corresponding EFs. In this case, an increase in computational complexity is anticipated.

There is also a desire to improve metaheuristics with the introduction of operations informed by the approach described in this study. Employing some aspects of the approach may provide a metaheuristic with a good indication of information that may be exploited in order to adapt itself or evolve more effective operators via hyperheuristics, which exploit the implied efficiency in the solution process. In summary, the approach is a fast, intuitively simple and practical tool for finding CCEFs for cardinality-constrained portfolio problems with potential to generalise to other problems. The approach forms a useful framework for decision-makers to exploit modern computational power at their disposal.

## Acknowledgements

The authors thank the Centre for Mathematical Sciences at the University of Plymouth for research support during the preparation of this work. Thanks also go to T. Alotaibi for translation of a preliminary version of the MAPLE code to the **R** language.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Wordcount

6042

## References

- Anagnostopoulos, K., & Mamanis, G. (2011). The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, 38(11), 14208–14217.
- Bach, F., Ahipaşaoğlu, S. D., & d’Aspremont, A. (2010). Convex relaxations for subset selection. *arXiv preprint arXiv:1006.3601*.
- Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11), 1069–1072.
- Bertsimas, D., & Cory-Wright, R. (2019). A scalable algorithm for sparse portfolio selection. *arXiv preprint arXiv:1811.00138*.



- Cesarone, F., Scozzari, A., & Tardella, F. (2010). Solutions to portfolio problems in beasley's or-library. Retrieved from [http://host.uniroma3.it/docenti/cesarone/datasetsw3\\_tardella.html](http://host.uniroma3.it/docenti/cesarone/datasetsw3_tardella.html) (accessed 08/04/19)
- Cesarone, F., Scozzari, A., & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, 205(1), 213–234.
- Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13), 1271–1302.
- Craven, M. J., & Graham, D. I. (2017). Exploring the (efficient) frontiers of portfolio optimization. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 19–20).
- Craven, M. J., & Graham, D. I. (2019). Parallel stochastic solution of portfolio optimisation problems with cardinality constraints. *submitted to SN Operations Research Forum*.
- Duran, F. E. C., Cotta, C., & Fernández-Leiva, A. J. (2012). A comparative study of multi-objective evolutionary algorithms to optimize the selection of investment portfolios with cardinality constraints. In *European conference on the applications of evolutionary computation* (pp. 165–173).
- Gao, J., & Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3), 745–761.
- Graham, D. I., & Craven, M. J. (2019). *Sifted qp solutions for portfolio optimisation problems dataset*. Retrieved from <http://hdl.handle.net/10026.1/14270>
- Lin, C.-C., & Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, 185(1), 393–404.
- Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24, 757–772.
- Mansini, R., & Speranza, M. G. (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, 114(2), 219–233.
- Michalewicz, Z., & Fogel, D. B. (2013). *How to solve it: modern heuristics*. Springer Science & Business Media.
- Moral-Escudero, R., Ruiz-Torrubiano, R., & Suárez, A. (2006). Selection of optimal investment portfolios with cardinality constraints. In *Evolutionary computation, 2006. cec 2006. ieee congress on* (pp. 2382–2388).